

Outputs for Backend

The screenshot displays the VS Code interface with a REST client request and response for a POST endpoint. The Explorer panel on the left shows the project structure for 'Backend FOS', including controllers, models, and routes. The main editor shows a REST client request to `http://127.0.0.1:3002/users/admin-login` with a POST method. The request body is a JSON object with the following fields:

```
{  "username": "Vaishnavi",  "email": "v@gmail.com",  "password": "123456",  "role": "admin"}
```

The response is a JSON object with the following fields:

```
{  "username": "Vaishnavi",  "password": "$2b$10$hq/1T6hs.5WyCVmf08Mo0u02pwe0bzqD9EZbaYko1NpgjN.qQmpKq",  "email": "v@gmail.com",  "role": "admin"}
```

The status of the response is 201 Created, with a time of 105 ms and a size of 449 B. The terminal panel at the bottom shows the output of the request, indicating a validation error:

```
[Symbol(mongoose:validatorError)]: true  
}_  
_message: 'Admin validation failed'
```

The Windows taskbar at the bottom shows the date and time as 10:28 PM on 11/4/2023.

Visual Studio Code interface showing a REST client request and response for the `admin-register` endpoint.

Request:

- Method: POST
- URL: `http://127.0.0.1:3002/admin/admin-register`
- Body (JSON):

```
{  "username": "Pratik",  "email": "p@gmail.com",  "password": "123456"}
```

Response:

- Status: 201 Created
- Time: 122 ms
- Size: 315 B
- Body (JSON):

```
{  "message": "Admin registered successfully"}
```

Terminal Output:

```
Successfully Connected
[nodemon] restarting due to changes...
[nodemon] starting 'node index.js'
Server is running on port 3002
Successfully Connected
```

Visual Studio Code interface showing a REST client request and response for the `customer/user-register` endpoint.

Request:

- Method: POST
- URL: `http://127.0.0.1:3002/customer/user-register`
- Body (JSON):

```
{  "username": "Vaishnavi",  "email": "vj@gmail.com",  "password": "123456"}
```

Response:

- Status: 201 Created
- Time: 104 ms
- Size: 318 B
- Body (JSON):

```
{  "message": "Customer registered successfully"}
```

Terminal Output:

```
at Object.<anonymous> (D:\LUU Projects MongoDB\Food Ordering Sys\Backend FOS\index.js:28:21)
Node.js v18.17.1
[nodemon] app crashed - waiting for file changes before starting...
[nodemon] restarting due to changes...
[nodemon] starting 'node index.js'
Server is running on port 3002
Successfully Connected
```

Backend FOS

EXPLORER

- BACKEND FOS
 - controllers
 - AdminController.js
 - AdminFoodController.js
 - OrderController.js
 - RestaurantController.js
 - UserController.js
 - middleware
 - authentication.js
 - model
 - Admin.js
 - FoodItem.js
 - FoodOrderingCust.js
 - Restaurant.js
 - UserModel.js
 - node_modules
 - route
 - admin.js
 - adminFoodRoute.js
 - customer.js
 - order.js
 - restaurant.js
 - index.js
 - package-lock.json
 - package.json
- OUTLINE
- TIMELINE

http://127.0.0.1:3002/admin-food-router/add-food-item

POST

Body

```
1 {
2   "name": "Pizza",
3   "quantity": 20,
4   "price": 100,
5   "description": "Italian Pizza"
6 }
```

Body

```
1 {
2   "message": "Food item added successfully"
3 }
```

PROBLEMS OUTPUT TERMINAL PORTS POSTMAN CONSOLE DEBUG CONSOLE

```
[nodeemon] starting 'node index.js'
Server is running on port 3002
Sucessfully Connected
```

Backend FOS

EXPLORER

- BACKEND FOS
 - controllers
 - AdminController.js
 - AdminFoodController.js
 - OrderController.js
 - RestaurantController.js
 - UserController.js
 - middleware
 - authentication.js
 - model
 - Admin.js
 - FoodItem.js
 - FoodOrderingCust.js
 - Restaurant.js
 - UserModel.js
 - node_modules
 - route
 - admin.js
 - adminFoodRoute.js
 - customer.js
 - order.js
 - restaurant.js
 - index.js
 - package-lock.json
 - package.json
- OUTLINE
- TIMELINE

http://127.0.0.1:3002/restaurant/restaurants

POST

Body

```
1 {
2   "name": "P&P",
3   "location": "Pune",
4   "description": "Pure Veg"
5 }
```

Body

```
1 {
2   "name": "P&P",
3   "description": "Pure Veg",
4   "location": "Pune",
5   "id": "6547147bacd4164b2040a65a"
6 }
```

PROBLEMS OUTPUT TERMINAL PORTS POSTMAN CONSOLE DEBUG CONSOLE

```
Sucessfully Connected
```

Visual Studio Code interface showing a REST client request for `http://127.0.0.1:3002/restaurant/restaurants`. The request is a POST method with a JSON body:

```
1 {
2   "name": "S&S",
3   "location": "Satara",
4   "description": "Non Veg"
5 }
```

The response status is 201 Created, Time: 17 ms, Size: 371 B. The response body is:

```
1 {
2   "name": "S&S",
3   "description": "Non Veg",
4   "location": "Satara",
5   "id": "654714a3acd4164b2040a65c".
```

The Explorer sidebar shows the project structure for `Backend FOS`, including controllers, models, middleware, authentication, and routes. The terminal shows "Successfully Connected".

Visual Studio Code interface showing a REST client request for `http://127.0.0.1:3002/restaurantorders/orders`. The request is a POST method with a JSON body:

```
5 {
6   "foodItem": "6547112a3e5c14f406326208",
7   "quantity": 3,
8   "price": 300.00
9 },
10 {
11   "foodItem": "6547118c3e5c14f40632620c",
12   "quantity": 3,
13   "price": 300.00
14 }
```

The response status is 201 Created, Time: 20 ms, Size: 645 B. The response body is:

```
1 {
2   "message": "Order created successfully",
3   "order": {
4     "customer": "65470f416d3f18fec355d9ed",
5     "orderItems": [
```

The Explorer sidebar shows the project structure for `Backend FOS`, including controllers, models, middleware, authentication, and routes. The terminal shows "Successfully Connected".

MongoDB Compass - localhost:27017/FoodOrderingSystem.orders

Connect Edit View Collection Help

localhost:27017 ... Documents FoodOrderingSys...

My Queries Databases Search

FoodOrderingSystem

- admins
- customers
- fooditems
- orders**
- restaurants

IMDB admin config insta local netflix

FoodOrderingSystem.orders

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#) Explain Reset Find Options

ADD DATA EXPORT DATA

1 - 2 of 2

```
{ "_id": ObjectId('65471679f453194494ed816e'),
  "customer": ObjectId('65470f026d3f18fec355d9ea'),
  "orderItems": Array (1)
    total: 100
    status: "pending"
    __v: 0
}
```

```
{ "_id": ObjectId('654716e8f453194494ed8171'),
  "customer": ObjectId('65470f416d3f18fec355d9ed'),
  "orderItems": Array (2)
    total: 600
    status: "pending"
    __v: 0
}
```

>_MONGOSH

Windows taskbar: 9:46 AM 11/5/2023

MongoDB Compass - localhost:27017/FoodOrderingSystem.fooditems

Connect Edit View Collection Help

localhost:27017 ... Documents FoodOrderingSys...

My Queries Databases Search

FoodOrderingSystem

- admins
- customers
- fooditems**
- orders
- restaurants

IMDB admin config insta local netflix

FoodOrderingSystem.fooditems

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#) Explain Reset Find Options

ADD DATA EXPORT DATA

1 - 4 of 4

```
{ "_id": ObjectId('6547112a3e5c14f406326288'),
  "name": "Pizza",
  "quantity": 20,
  "price": 100,
  "description": "Italian Pizza",
  "__v": 0
}
```

```
{ "_id": ObjectId('6547117d3e5c14f40632628a'),
  "name": "Pasta",
  "quantity": 20,
  "price": 100,
  "description": "Italian Pasta",
  "__v": 0
}
```

```
{ "_id": ObjectId('6547118c3e5c14f40632628c'),
  "name": "Pasta"
}
```

>_MONGOSH

Windows taskbar: 9:46 AM 11/5/2023

MongoDB Compass - localhost:27017/FoodOrderingSystem.admins

Connect Edit View Collection Help

localhost:27017 ... Documents FoodOrderingSys...

My Queries Databases Search

FoodOrderingSystem

- admins
- customers
- fooditems
- orders
- restaurants

IMDB admin config insta local netflix

FoodOrderingSystem.admins

2 DOCUMENTS 3 INDEXES

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#)

EXPLAIN RESET FIND Options

ADD DATA EXPORT DATA

1 - 2 of 2

```
{ "_id": ObjectId("654677e67954c7dfdc7d52f"), "username": "Vaishnavi", "password": "$2b$10$Shq/1T6hs.5MyCVmf08Mo0u02pwe0bzqD9EZbaYko1NpgjN.qQmpXq", "email": "v@gmail.com", "role": "admin", "__v": 0 }
```

```
{ "_id": ObjectId("6546858811e5db328dce02dc"), "username": "Pratik", "email": "p@gmail.com", "password": "$2b$10$EH1Uqe20H6xgJN2Vg0Rpa.Hnatu1q6drNGyiyAzvUs6HLTRS6/28e", "__v": 0 }
```

> MONGODB

MongoDB Compass - localhost:27017/FoodOrderingSystem.customers

Connect Edit View Collection Help

localhost:27017 ... Documents FoodOrderingSys...

My Queries Databases Search

FoodOrderingSystem

- admins
- customers
- fooditems
- orders
- restaurants

IMDB admin config insta local netflix

FoodOrderingSystem.customers

0 DOCUMENTS 3 INDEXES

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#)

EXPLAIN RESET FIND Options

ADD DATA EXPORT DATA

1 - 2 of 2

```
{ "_id": ObjectId("65470f026d3f18fec355d9ea"), "username": "Dnyana", "email": "d@gmail.com", "password": "$2b$10$L/3pOUem/Tcku.CwehCINeXdtqPc75F5ehKVltbX.FgEZijRvS.", "__v": 0 }
```

```
{ "_id": ObjectId("65470f416d3f18fec355d9ed"), "username": "Vaishnavi3", "email": "vj@gmail.com", "password": "$2b$10$CrktsBH3CdRu.6NgEaYEu.vUSG67KAjmcQorp2EoNQlt601xDqAu", "__v": 0 }
```

> MONGODB

MongoDB Compass - localhost:27017/FoodOrderingSystem.restaurants

Connect Edit View Collection Help

localhost:27017 Documents FoodOrderingSys...

My Queries Databases Search

FoodOrderingSystem

- admins
- customers
- fooditems
- orders
- restaurants**
- IMDB
- admin
- config
- insta
- local
- netflix

FoodOrderingSystem.restaurants

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#) Explain Reset Find Options

ADD DATA EXPORT DATA 1 - 2 of 2

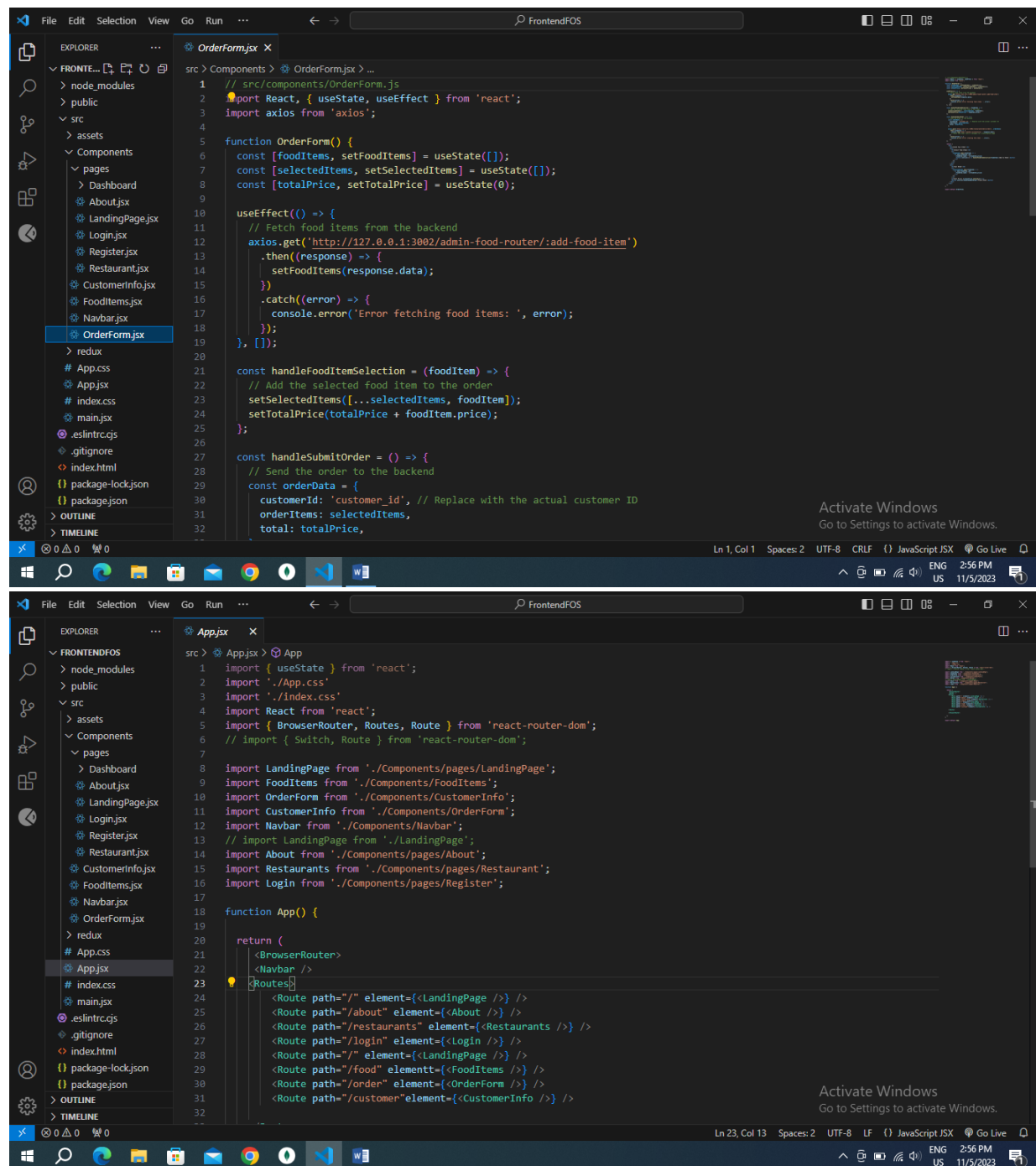
```
{ "_id": ObjectId("6547147bacd4164b2040a65a"),  
  "name": "P&P",  
  "description": "Pure Veg",  
  "location": "Pune",  
  "__v": 0  
}
```

```
{ "_id": ObjectId("654714a3acd4164b2040a65c"),  
  "name": "S&S",  
  "description": "Non Veg",  
  "location": "Satara",  
  "__v": 0  
}
```

>_MONGOOSH

Windows taskbar: 9:46 AM, 11/5/2023

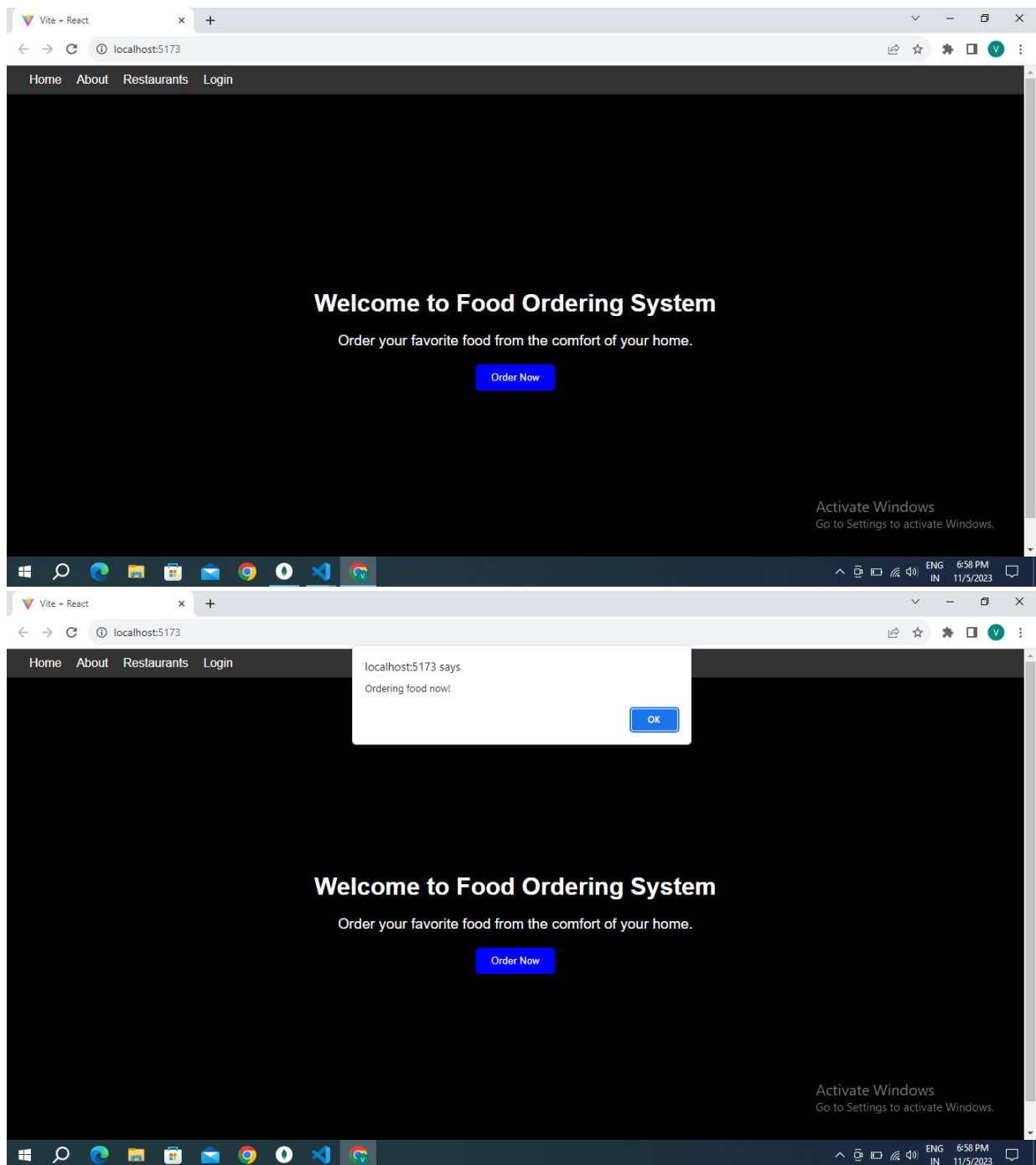
Code For Frontend




```
File Edit Selection View Go Run ... FrontendFOS
EXPLORER
  FRONTENDFOS
    > node_modules
    > public
    > src
      > assets
      > Components
        > pages
          > Dashboard
            About.jsx
            LandingPage.jsx
            Login.jsx
            Register.jsx
            Restaurant.jsx
            CustomerInfo.jsx
            FoodItems.jsx
            Navbar.jsx
            OrderForm.jsx
          > redux
            # App.css
            App.jsx
            # index.css
            main.jsx
            .eslintrc.js
            .gitignore
            package-lock.json
            package.json
        > OUTLINE
        > TIMELINE
src > Components > pages > Register.jsx > ...
1  /* eslint-disable react/no-unescaped-entities */
2  import { Link } from 'react-router-dom';
3  import { useState } from 'react';
4  import { useDispatch } from 'react-redux';
5  // import { registerUser } from '../redux/feature/authSlice';
6  import { useNavigate } from 'react-router-dom';
7
8  const Register = () => {
9    const dispatch = useDispatch();
10    const navigate = useNavigate();
11    const [email, setEmail] = useState('');
12    const [password, setPassword] = useState('');
13
14    const handleRegister = (e) => {
15      e.preventDefault();
16      dispatch(registerUser({ email, password, role: 'user' }));
17      navigate('/dashboard');
18    }
19
20    return (
21      <section className='login-section'>
22        <div>
23          <h1>Sign Up</h1>
24          <form onSubmit={handleRegister}>
25            <div>
26              <label htmlFor='email'>Email</label>
27              <input type='email' value={email} onChange={e => setEmail(e.target.value)} id='email' placeholder='>
28            </div>
29            <div>
30              <label htmlFor='password'>Password</label>
31              <input type='password' value={password} onChange={e => setPassword(e.target.value)} id='password' placeholder='>
32            </div>
33          </form>
34        </div>
35      </section>
36    );
37  }
38
39  export default Register;
```

```
File Edit Selection View Go Run ... FrontendFOS
EXPLORER
  FRONTENDFOS
    > node_modules
    > public
    > src
      > assets
      > Components
        > Dashboard
          AdminDashboard.jsx
          Dashboard.jsx
          UserDashboard.jsx
          About.jsx
          LandingPage.jsx
          Login.jsx
          Register.jsx
          Restaurant.jsx
          CustomerInfo.jsx
          FoodItems.jsx
          Navbar.jsx
          OrderForm.jsx
        > redux
          # App.css
          App.jsx
          # index.css
          main.jsx
          .eslintrc.js
          .gitignore
          package-lock.json
          package.json
        > OUTLINE
        > TIMELINE
src > Components > Dashboard > AdminDashboard.jsx > ...
1  const AdminDashboard = () => {
2    return (
3      <section style={{ background: 'rgba(0,0,0,0.5)', height: '100vh' }}>
4        <form>
5          <div>
6            <label>Name</label>
7            <input type='text' name='' id='' />
8          </div>
9          <div>
10             <label>Description</label>
11             <input type='text' name='' id='' />
12           </div>
13           <div>
14             <label>Genre</label>
15             <input type='text' name='' id='' />
16           </div>
17           <div>
18             <label>Duration</label>
19             <input type='text' name='' id='' />
20           </div>
21           <div>
22             <label>Video URL</label>
23             <input type='text' name='' id='' />
24           </div>
25         </form>
26       </section>
27     );
28   }
29
30   export default AdminDashboard;
```

Outputs of website



Sign In

Email

Password

Sign In

☐ Remember me

Don't have account [Sign up](#)

Activate Windows
Go to Settings to activate Windows.

Sign Up

Email

Email

Password

Sign Up

Already have account [Sign in](#)

Activate Windows
Go to Settings to activate Windows.

